REMARKS

Introduction

The Office Action dated February 10, 2005, and the references cited therein have been carefully considered. In response, the applicant offers the following remarks.

Office Action Objection

Claim 10 was objected to due to grammatical errors. In response, claim 10 is currently amended to correct the grammatical error.

Office Action Rejection

Claims 1-22 and 24 are pending and were rejected under 35 USC 102(e) in view of Jackowski, U.S. patent 6,141,686 (" '686 patent").

The applicant notes that U.S. patent no. 6,167,445 to Gai et al is listed in the Notice of References Cited but is not mentioned in the Office Action text.

Summary of the Prior Art

The '686 patent discloses a system of low level network service-provider plugins. The plugins are controlled by an extensible service provider that is layered above the TCP or other protocol layer but below the Winsock-2 library and API. The system has policy servers that determine priority of network traffic through control points on a network.

Arguments for Patentability

In order for a claim to be anticipated under §102, the anticipating references must disclose at least one embodiment that incorporates all the claimed elements. *In re Bond*, 15 USPQ2d 1566, 1567 (Fed. Cir. 1990) ("For a prior art reference to anticipate in terms of 35 USC §102, every element of the claimed invention must be identically shown in a single reference.... These elements must be arranged as in the claims under review....".

All the pending claims call for detection of transmission or receipt of data at the transport layer. For example, claim 1 begins:

> A method of tracing data traffic on a network, the method comprising:
> at the transport layer of a protocol stack...

The '686 patent does not disclose detection at the transport layer as called for in all the independent claims. The detection in the '686 patent occurs "between Winsock-2 library 34 and TCP layer 40." Col. 8, lines 12-13. "Between" is not the same as "at."

The way the system in the '686 patent operates requires that it sit above the TCP layer because it also monitors high level applications. Col. 7, lines 34-44. Furthermore, the '686 patent explicitly criticizes a low level, TCP based system as claimed. The '686 patent states:

> The inventors have realized that policy servers typically
> prioritize network traffic based on low-level IP addresses and
> TCP ports. Low-level prioritization is undesirable because IP
> addresses and TCP ports are often shared by many applications
> and users. All applications and users sharing IP addresses and
> ports would be given the same priority. Even when IP
> addresses are statically assigned to one machine, all
> applications on that machine may need to be given the same
> priority, even though some applications are inherently more
> important than others. (col. 6, lines 58-67)

In regard to claim 1, the Office Action refers to col. 9, lines 9-12 for disclosing "at the transport layer of a protocol stack residing on a first device in the network." Col. 9, lines 9-12 states:

> Winsock-2 library 42 receives calls from DCOM server 64 that
> send and receive data packets over the network, passing down
> the protocol stack through TCP layer 40 and IP layer 41.

This quote does not disclose detection at the transport layer as called for in claim 1. It describes data flow through the various layers of the OSI model without disclosing detection at the transport layer as claimed.

In regard to independent claim 10, the Office Action points to col. 10, lines 55-67 as disclosing detection of transmitted or received data at the transport layer. Col. 10, lines 55-67 states:

> FIG. 7A shows the EventAppInit data-object definition. A version number is the first parameter in the definition. A timestamp is stored for the time that the event activated the application-classifier plugin. The unique process ID for the Winsock process that originated the event is stored in the ProcessID field. One application can spawn several network processes, but each process is identified by a unique ID assigned by the Windows operating system.
>
> The name of the high-level application is retrieved using the Win32 API library call GetModuleFileName. Thus the objects based on EventAppInit data-object definition include the names of the application, user, and host, as well as a timestamp and the unique process ID.

Again, there is no disclosure of detection occurring at the transport layer just a discussion of data flows.

In regard to independent claim 12, the Office Action points to col. 9, lines 9-12 to disclose detection at the transport layer. As discussed in reference to claim 1, this disclosure does not anticipate claim 12 because there is no mention of detection at the transport layer as claimed in claim 12. As the independent claims are not anticipated, the dependent claims also are not anticipated.

Accordingly, the '686 patent does not disclose using the transport layer to monitor data flows as called for in the independent claims. In fact, the '686 patent explicitly reasons why it does not use the transport layer to monitor traffic. Accordingly, all the elements of the pending claims are not present in the '686 patent and the pending claims should be allowed.

If the Examiner has any questions, the Examiner is encouraged to call the undersigned at (312) 474-6300. Applicant believes no fee is due with this response. However, if a fee is due, please charge our Deposit Account No. 13-2855, under Order No. 30835/39086 from which the undersigned is authorized to draw.

Dated: May 10, 2005   Respectfully submitted,

                                               By _W.J. Kramer_
                                               William J. Kramer
                                                  Registration No.: 46,229
                                               MARSHALL, GERSTEIN & BORUN LLP
                                               233 S. Wacker Drive, Suite 6300
                                               Sears Tower
                                               Chicago, Illinois 60606-6357
                                               (312) 474-6300
                                               Attorney for Applicant